



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Metody wirtualizacji [S1Cybez1>MW]

Przedmiot

Kierunek studiów

Cyberbezpieczeństwo

Rok/Semestr

2/3

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

8

Laboratorium

24

Inne

0

Ćwiczenia

0

Projekty/seminaria

24

Liczba punktów ECTS

4,00

Koordynatorzy

dr inż. Łukasz Kułacz

lukasz.kulacz@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student powinien posiadać podstawową wiedzę z zakresu programowania, w szczególności pojęcia zmiennych, klas i funkcji. Powinien posiadać umiejętność implementacji prostych programów i aplikacji sieciowych.

Cel przedmiotu

Celem przedmiotu jest przekazanie studentom podstawowej wiedzy dotyczącej wirtualizacji i konteneryzacji. Poprzez wirtualizację student może odizolować projektowaną aplikację lub usługę od systemu operacyjnego hosta, dzięki czemu możliwe jest zapewnienie niezależnego od sprzętu, wersji systemu i sterowników działania kodu. Dodatkowo w ramach przedmiotu przedstawione zostaną zagadnienia związane z konteneryzacją, przede wszystkim, w kontekście skalowalności i niezależności aplikacji projektowanych jako tzw. mikrousługi.

Przedmiotowe efekty uczenia się

Wiedza:

Student ma podstawową wiedzę teoretyczną i praktyczną w zakresie wirtualizacji oprogramowania, konteneryzacji i orkiestracji kontenerów. Student zna podstawowe narzędzia umożliwiające tworzenie i

używanie kontenerów, a także zarządzanie nimi. Student zna cechy i ograniczenia wynikające z zastosowania kontenerów do wdrażania gotowego rozwiązania.[K1_W06][K1_W11]

Umiejętności:

Student potrafi tworzyć wirtualne systemy operacyjne, korzystać z nich i nimi zarządzać. Student potrafi przygotować proste aplikacje i następnie umieszczać je w kontenerach, uruchamiać i aktualizować. Student potrafi łączyć kilka mikrousług w działający system i podmieniać elementy systemu bez konieczności uruchamiania całego systemu od nowa. [K1_U02]

Kompetencje społeczne:

Student ma świadomość możliwości i ograniczeń związanych z tworzeniem oprogramowania i umieszczaniem go wewnątrz kontenerów. Rozumie potencjalne zyski związane ze skalowalnością i zwiększoną niezawodnością systemu jednocześnie dostrzegając ryzyka związane z zastosowaniem takiego rozwiązania.[K1_K05]

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

W zakresie projektu weryfikowanie założonych efektów kształcenia realizowane jest przez: ocenę merytoryczną realizacji projektu indywidualnego.

W zakresie laboratorium weryfikowanie założonych efektów kształcenia realizowane jest przez: ocenę merytoryczną zadań indywidualnych lub grupowych wykonywanych w ramach zajęć lub w postaci zadań do wykonania po zajęciach, aktywność na zajęciach.

W zakresie wykładu weryfikowanie założonych efektów kształcenia realizowane jest poprzez pisemne zaliczenie w charakterze testu, który może zawierać pytania jednokrotnego, wielokrotnego wyboru oraz pytania otwarte.

W każdej formie zaliczenia przedmiotu ocena zależy od liczby zdobytych przez studenta punktów w stosunku do maksymalnej liczby punktów obowiązkowych. Warunkiem pozytywnego zaliczenia jest otrzymanie co najmniej 50% punktów możliwych do zdobycia. Zależność oceny od liczby punktów definiuje Regulamin Studiów. Dodatkowo zasady zaliczania przedmiotu i dokładne progi zaliczeniowe zostaną przekazane studentom na początku semestru z wykorzystaniem uczelnianych systemów elektronicznych oraz na pierwszych zajęciach (w każdej formie zajęć).

Treści programowe

1. Witualizacja systemów operacyjnych.
2. Podstawy konteneryzacji.
3. Podstawy mikrousług.
4. Podstawy orkiestracji kontenerów.
5. Integracja mikrousług.

Tematyka zajęć

1. Witualizacja systemów operacyjnych.
2. Podstawy konteneryzacji.
3. Podstawy mikrousług.
4. Podstawy orkiestracji kontenerów.
5. Integracja mikrousług.

Metody dydaktyczne

Wykład: prezentacja multimedialna, uzupełniana przykładami i dodatkowymi wyjaśnieniami na podstawie fragmentów kodu.

Laboratoria: praca przy komputerze, wykonanie indywidualnych zadań, wykonanie grupowych zadań.

Projekt: praca przy komputerze, wykonanie indywidualnego zadania.

Literatura

Podstawowa:

1. Rozwijanie mikrousług w Pythonie. Budowa, testowanie, instalacja i skalowanie. Tarek Ziade

2. Architektura aplikacji w Pythonie. TDD, DDD i rozwój mikrousług reaktywnych. Harry Percival
3. Nauka Dockera w miesiąc. Elton Stoneman
4. Nauka Kubernetesa w miesiąc. Elton Stoneman

Uzupełniająca:

1. Docker. Niezawodne kontenery produkcyjne. Praktyczne zastosowania. Sean Kane

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	116	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	56	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwίων/egzaminu, wykonanie projektu)	60	2,00